(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.		(11) 공개번호	특1999-0076101
G06F 17/60		(43) 공개일자	1999년10월15일
(21) 출원번호	10-1998-0010767		
(22) 출원일자	1998년03월27일		

(71) 출원인 한국 정보 보호 센터, 이재우
대한민국
137070
서울특별시 서초구 서초동 1321-6

(72) 발명자 이재우
대한민국
140-240
서울특별시 용산구 서빙고동 214(20/1) 신동아아파트 2-401

이홍섭
대한민국
136-060
서울특별시 성북구 돈암동 616-100 한신아파트 101-1210

고승철
대한민국
411-370
경기도 고양시 일산구 주엽동49 강선마을 506-1801

박정호
대한민국
156-070
서울특별시 동작구 흑석동 명수대 현대아파트 101-1210

이영철
대한민국
139-208
서울특별시 노원구 상계8동 주공아파트 1014-1304

김기현
대한민국
705-036
대구광역시 남구 대명6동 604-9 (10/3)

은유진
대한민국
449-900
경기도 용인시 기흥읍 구갈리 385-1 한성2차아파트 204-204

정현철
대한민국
142-104
서울특별시 강북구 미아4동 3-109

이정효
대한민국
152-056
서울특별시 구로구 구로6동 1257번지 럭키아파트2-707

권석철
대한민국
110-260
서울특별시 종로구 가회동 22번지

(74) 대리인 박해천
원석희

(77) 심사청구 있음

(54) 출원명 패스워드 교환방식을 이용한 사용자-서버간의 상호 신분 인증방법

요약

1. 청구범위에 기재된 발명이 속한 분야

본 발명은 패스워드 교환방식을 이용한 사용자-서버간의 상호 신분 인증 방법에 관한 것임.

2. 발명이 해결하려고 하는 기술적 과제

본 발명은 안전하게 패스워드를 교환함으로써 패스워드 누출, 패스워드 도용 및 신분 위장을 방지하고, 통신로상에서 인증관련 데이터의 교환 횟수를 감소시킬 수 있는 사용자-서버간의 상호 신분 인증 방법을 제공하는데 그 목적이 있음.

3. 발명의 해결 방법의 요지

본 발명은, 사용자와 서버간의 데이터를 초기화하는 제1 단계; 사용자가 접근 요구하는 제2 단계; 서버(Server)가 상기 접근 요구를 수신하고, 도전값($N \parallel R^{\oplus} X_{N+1} \parallel E_R(X_{N+1})$)을 생성하여 사용자에게 전송하는 제3 단계; 서버를 인증하는 제4 단계; 암호화된 해쉬값($E_{R'}(H_N(P))$)을 상기 서버에게 응답값으로 전송하는 제5 단계; 및 상기 응답값 $E_{R'}(H_N(P))$을 수신하여 사용자 신분을 확인하는 제6 단계를 포함함.

4. 발명의 중요한 용도

컴퓨터 통신 분야, 전자상거래 분야

대표도

도4

명세서

도면의 간단한 설명

도1은 인증이 요구되는 일반적인 컴퓨터 시스템의 블록도.

도2는 사용자와 서버(server)간의 기본적인 정보 교환 과정의 흐름도.

도3은 본 발명에 따른 초기화 과정의 일실시예 흐름도.

도4는 본 발명에 따른 인증 과정의 일실시예 상세 흐름도.

도5는 본 발명에 따른 인증 과정의 일실시예 상세 흐름도.

* 도면의 주요 부분에 대한 부호 설명

11: 사용자 컴퓨터 12: 서버 컴퓨터

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 사용자-서버간의 상호 신분 인증 방법에 관한 것으로, 특히 안전하게 패스워드를 교환함으로써 정보를 주고 받는 실체들간에 서로에 대한 신분을 인증하여, 패스워드 누출 및 도용, 신분위장으로부터 상호 신뢰성을 확보하도록 하는 사용자-서버간의 상호 신분 인증 방법에 관한 것이다.

오늘날 세계 각국의 통신망이 상호 연결되어 인터넷을 통한 정보의 교환이 일반화되어가고 있는 반면에 개인정보의 누출, 전산망 해킹(hacking) 등 그 역기능 현상 또한 심각한 사회문제로 대두되고 있다. 그 동안 국내에서 발생한 대표적인 해킹 행위는 주로 단순 침입, 사용자 식별번호(ID) 도용, 자료 절취, 자료 변조 및 파괴 등이며 외국의 해킹 실태는 국내에 비하여 매우 심각한 실정이다.

사용자 식별번호(ID) 및 패스워드(password)를 인증 기반으로 하고 있는 현재의 유닉스(UNIX) 시스템에서 패스워드의 누출은 많은 위험성을 내포하고 있다. 최근 네트워크를 감청하거나 식별번호(ID)와 패스워드를 도용하는 해킹 방법이 많이 사용되고 있다. 또한 국내에서 발생한 해킹 사례의 많은 부분들이 타인의 식별번호(ID)와 패스워드를 도용하거나 또는 이를 이용하여 해킹하는 사례가 주류를 이루고 있다. 1996년 9월의 경우에 인터넷 서비스 망에 홈뱅킹 이용자의 패스워드와 식별번호(ID)를 가로챌 수 있는 변형 원격접속 프로그램을 설치하여 계좌이체를 시도하다 검거된 사례도 발생하였다.

이처럼 패스워드 도용을 이용한 불법 접속 시도 등 각종 위협에서 전산망을 안전하게 운용하기 위하여 사용자의 패스워드 누출 및 도용 방지, 사용자 신분 위장 및 불법 시도를 방지할 수 있을 뿐만 아니라 PC 통신망, 금융망 등에 쉽게 적용할 수 있는 전산망 원격 사용자 인증 기술이 요구된다.

현재 사용되고 있는 인증 기술의 일예인 S/KEY 일회용 패스워드 시스템은 미국의 벨코어(Bellcore)사에서 만든 제품으로, 통신로상의 도청이나 재시도 공격으로부터 안전한 사용자 인증기능을 제공한다. 이 시스템은 기존의 일회용 또는 다목적 인증 시스템들에 비해 여러 가지 장점을 가지고 있다. 사용자의 비밀 패스워드는 결코 네트워크 상을 지나지 않는다. 어떤 비밀 정보도 저장되어지지 않으며, 하부 알고리즘은 공개된 지식이며, 이 시스템의 원격지에서는 어떤 컴퓨터라도 이용가능하다. 또한, S/KEY 인증 방식은 사용자의 패스워드를 보호하기 위한 간단한 스킴이다. 이 제품은 거의 모든 유닉스(UNIX)시스템에 추가적인 하드웨어없이, 패스워드 정보를 저장하지 않고 쉽고 빠르게 설치할 수 있다.

S/KEY 일회용 패스워드 인증 방식은 두가지 측면이 존재한다. 하나는 사용자 또는 클라이언트(client) 측면인데, 적절한 일회용 패스워드가 생성되어져야 한다. 다른 하나는 시스템 또는 서버(Server) 측면으로, 수신된 일회용 패스워드가 적절한 것인지 검사되어져야 한다. 일회용 패스워드는 단방향 해쉬함수(One-Way Hash Function)를 이용해서 생성되고 검사되어진다는 장점이 있다.

해쉬함수는 일방향으로 계산이 빠르고 용이하게 되어지지만, 역방향으로는 계산이 이론적으로 불가능한 함수를 말한다. 즉, 해쉬함수 H()가 입력값 x를 입력으로 받아 출력값 y를 생성하게 되는 것은 빠르고 용이하지만, 출력 값 y로부터 입력값 x를 계산해내는 것은 이론적으로 매우 어려운 함수이다.

S/KEY의 일회용 패스워드는 이러한 성질을 가진 단방향 해쉬함수를 여러번 적용함으로써 계속해서 생성된다. 즉, 첫 번째 일회용 패스워드는 사용자의 비밀키(S)를 소정의 수 n만큼의 해쉬함수를 수행함으로써 생성된다. 그리고 서버(server)측에는 사용자 비밀키(s)를 n+1번 해쉬함수를 수행한 값으로 저장하도록 한다. 만약 n=4라고 가정하면, 첫 번째 패스워드 p(1)은 다음과 같이 생성된다.

p(1)=H(H(H(H(s))))

두 번째 일회용 패스워드는 사용자의 패스워드를 단방향 함수에 n-1번, 즉 3번 수행함으로 생성되어진다.

p(2)=H(H(H(s))))

일회용 패스워드 p(i)의 사용을 모니터하고 있는 도청자는 해쉬함수의 특성 상 다음 패스워드 p(i+1)을 생성해낼 수 없을 것이다. 처음 시점에서 사용자의 비밀키 (s)를 알지 못하면 도청이 불가능하게 된다.

처음에 서버(server) 컴퓨터는 수신한 일회용 패스워드의 복사본을 저장하고, 그것을 해쉬함수에 적용한다. 만약 그 결과가 시스템의 패스워드 파일안에 저장되어 있던 복사본과 일치하지 않으면, 그 인증 요구는 실패하게 된다. 만약 그들이 일치하면, 시스템 패스워드 파일안에 있는 사용자의 정보는 단방향 해쉬함수의 마지막 실행전에 저장되어 있던 일회용 패스워드의 복사본으로 갱신된다.

하지만, S/KEY 일회용 패스워드 방식의 취약점은 모든 정보가 평문(Plaintext)으로 전파된다는 것이다. 이것은 도전(challenge)값과 응답(response)값을 알 수 있다는 것을 의미하며, 이 정보들을 가지고 사전의 단어들에 적용한 도전(challenge)값의 결과와 계속해서 비교하면, 어느 순간 일치하는 사용자의 패스워드를 알아낼 수 있게 된다는 문제점이 있다.

이것은 현재 S/KEY 소프트웨어가 클라이언트(client)나 서버 어디서도 사용자 패스워드 선택시에 보안성 검사를 하지 않는다는 중요한 결점을 가지고 있다.

또한, 사용자가 사용하는 해쉬함수의 횟수가 서버로부터 사용자 측으로 전송되기 때문에, 도용자가 서버(server)로 위장하여 공격할 수 있다는 치명적인 문제점이 있다. 이 방법은 가짜 게이트웨이(Gateway)를 설정함으로 수행될 수 있다. 이하에 실예를 들어 공격 방법을 설명한다.

사용자가 위장된 가짜 서버(server)에 접속을 시도하면 원래의 98번째의 도전(challenge) 값을 받아야 하는데 대신에 55번째의 도전(challenge) 값을 받게 된다. 사용자는 그의 패스워드를 이용하여 55번째에 대한 응답(response)값을 생성한다. 위장된 가짜 서버는 로그인이 틀렸다고 말하고, 사용자가 실제로 원하는 서버로 다음 연결을 하게 한다. 여기서 55번째 도전(challenge)값으로부터 얻어진 응답(response) 값을 가지고 공격자는 해쉬 함수를 이용하여 나머지 응답(response) 값들을 알아낼 수 있게 된다. 예를들면, 지금 가지고 있는 정보를 이용해서 60번째 도전값에 대한 응답(response) 값을 알고 싶다면, 해쉬함수를 다섯 번만 수행하면 원하는 응답(response)값을 얻을 수 있게 된다.

발명이 이루고자 하는 기술적 과제

상기 문제점을 해결하기 위하여 안출된 본 발명은, 안전하게 패스워드를 교환함으로써 패스워드 누출, 패스워드 도용 및 신분 위장을 방지하고, 통신로상에서 인증관련 데이터의 교환 횟수를 감소시킬 수 있는 사용자-서버간의 상호 신분 인증 방법을 제공하는데 그 목적이 있다.

발명의 구성 및 작용

상기 목적을 달성하기 위한 본 발명의 방법은, 사용자와 서버간의 상호 신분 인증 방법에 있어서, 사용자와 서버가 초기화하여 상기 서버가 해쉬함수 수행 횟수인 N+1(여기서, N은 자연수) 및 해쉬 결과값 X_{N+1} 을 저장하는 제1 단계; 상기 사용자가 접근 요구를 전송하는 제2 단계; 상기 서버(Server)가 상기 접근 요구를 수신하여, 난수 R을 생성하고, 해쉬 수행 횟수 N, 난수 R과 상기 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값 및 상기 저장된 정보 X_{N+1} 의 암호화 값을 연쇄시킨 도전값($N \parallel R \oplus X_{N+1} \parallel E_R(X_{N+1})$)을 생성하여 사용자에게 전송하는 제3 단계; 상기 도전값을 수신한 상기 사용자가 자신의 비밀키 P와 해쉬함수를 이용하여 서버를 인증하기 위한 정보 $H_N(P)$, $H_{N+1}(P)$를 계산한후, $H_{N+1}(P)$과 상기 도전값 중 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값($R \oplus X_{N+1}$)의 배타적 논리합 연산을 수행하여 그 결과값 $R'(=(H_{N+1}(P) \oplus (R \oplus X_{N+1})))$를 구하고, 상기 R' 및 상기 도전값중 상기 저장된 정보 X_{N+1} 의 암호화 값 $E_R(X_{N+1})$을 복호화한 값 $D_{R'}(E_R(X_{N+1}))$을 $H_{N+1}(P)$ 값과 비교하여 일치하는지 여부를 판단하여 서버를 인증하는 제4 단계; 상기 $H_N(P)$을 상기 R'로 암호화하여 암호화값($E_{R'}(H_N(P))$을 상기 서버에게 응답값으로 전송하는 제5 단계; 및 상기 서버는 상기 응답값 $E_{R'}(H_N(P))$을 수신한후, 상기 난수 R을 이용하여 복호화하고 복호화된 결과값을 해쉬함수의 입력으로하여 계산하여 결과를 상기 저장된 값 X_{N+1} 과 비교하여 사용자 신분을 확인하는 제6 단계를 포함한다.

또한, 상기 목적을 달성하기 위한 본 발명의 다른 방법은, 사용자와 서버간의 상호 신분 인증 방법에 있어서, 사용자와 서버가 초기화하여 상기 서버가 해쉬함수 수행 횟수인 N+1(여기서, N은 자연수) 및 해쉬 결과값 X_{N+1} 을 저장하는 제1 단계; 상기 사용자가 접근 요구를 전송하는 제2 단계; 상기 서버(Server)가 상기 접근 요구를 수신하여, 난수 R을 생성하고, 해쉬 수행 횟수 N, 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값 및 상기 난수 R과 상기 저장된 정보 X_{N+1} 의 연쇄값을 해쉬한 값을 연쇄시킨 도전값($N \parallel R \oplus X_{N+1} \parallel H(R \parallel X_{N+1})$)을 생성하여 사용자에게 전송하는 제3 단계; 상기 도전값을 수신한 상기 사용자가 자신의 비밀키 P와 해쉬함수를 이용하여 서버를 인증하기 위한 정보 $H_N(P)$, $H_{N+1}(P)$를 계산한후, $H_{N+1}(P)$과 상기 도전값 중 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값 ($R \oplus X_{N+1}$)의 배타적 논리합 연산을 수행하여 그 결과값 $R'(=(H_{N+1}(P) \oplus (R \oplus X_{N+1})))$를 구하고, 상기 R'과 상기 $H_{N+1}(P)$를 해쉬한 값 $H(R' \parallel H_{N+1}(P))$을 $H(R \parallel X_{N+1})$ 값과 비교하여 일치하는지 여부를 판단하여 서버를 인증하는 제4 단계; 상기 $H_N(P)$을 상기 R'과 논리합 연산하여 결과값($H_N(P) \oplus R'$)인 응답값을 상기 서버에게 전송하는 제5 단계; 및 상기 서버는 상기 응답값 ($H_N(P) \oplus R'$)을 수신한후, 상기 난수 R과 상기 응답값을 논리합 연산하여 해쉬한 값($H(R \oplus (H_N(P) \oplus R'))$)이 상기 저장된 값 X_{N+1} 과 일치하는지 여부를 비교하여 사용자 신분을 확인하는 제6 단계를 포함한다.

이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예들을 상세히 설명한다.

도1은 인증이 요구되는 일반적인 컴퓨터 시스템의 블록도이다.

사용자는 사용자의 컴퓨터 시스템(11)에서 서비스를 제공하는 서버(server) 컴퓨터(12)에 접속하여 원하는 서비스를 제공받고자 한다. 이때, 원하는 서비스를 제공하는 측면이나 받는 측면 양측 모두에게 서로에 대한 신뢰성이 요구된다. 상호 인증은 사용자 컴퓨터(11)와 서비스 제공 서버 컴퓨터(12)간에 신뢰성을 제공하고자 하는 것이다.

도2는 사용자와 서버(server)간의 기본적인 정보 교환 과정의 흐름도이다.

사용자가 해당 서버(server)에 접속을 요구하면(21), 서버측에서는 난수인 도전(challenge)값을 사용자측에 전송한다(22). 사용자측은 수신한 상기 도전값으로 서버를 인증한후(23), 인증에 성공하면 사용자측에서 응답(response) 값을 서버측에 보낸다(24). 응답 값을 수신한 서버는 그 값을 이용하여 사용자를 인증한다(25). 상호 인증이 완료되면 인증 과정을 종료하고, 양측 모두가 서로를 신뢰하고 상호 인증을 기반으로 통신을 개시한다. 단계 23 또는 25, 즉 서버 인증 또는 사용자 인증이 실패하면 단계 21로 다시 돌아간다.

도3은 본 발명에 따른 초기화 과정의 일실시예 흐름도이다.

본 발명은 초기에 사용자측과 서버(server)측 상호간에 일련의 협의과정인 초기화 과정이 필요하다. 초기화 과정에서 사용자와 서버(server)는 해쉬함수를 적용할 횟수 N(여기서, N은 자연수)을 결정한다. 또한, 소정의 자연수 양수 N을 이용해서 사용자의 비밀정보 즉 비밀키와 해쉬함수를 이용하여 서버측에 저장할 정보를 생성해낸다. 해쉬함수를 H(), 사용자 비밀키를 P, 소정의 자연수를 N이라 할 때, 다음과 같이 계산해 낼 수 있다.

$H(P)=X_1$

$H(X_1)=X_2$

$H(X_2)=X_3$

- - -

$H(X_{N-1})=X_N$

$H(X_N)=X_{N+1}$

이 계산의 결과로 생성된 정보 X_{N+1} 과 N+1만을 서버측에 저장하고 사용자의 비밀키 P는 저장하지 않는다. 이렇게 함으로써 사용자의 비밀키는 사용자의 머리속에만 남게되다. 이는 기존의 유닉스(UNIX) 인증 시스템의 패스워드 파일에 대한 취약점을 제거하게 되는 것이다. 일련의 과정을 마치고 나면 본 발명의 초기화 과정은 종료된다.

도4는 본 발명에 따른 인증 과정의 일실시예 상세 흐름도이다.

앞에서 설명한 초기화 과정을 이용해서 도4에 도시된 것과 같은 사용자와 서버간의 상호 인증 절차가 수행된다.

먼저, 사용자가 사용자 식별번호(ID)를 가지고 접근 요구를 서버측에 보낸다(41). 서버측에서는 난수 R을 생성한 후(42), 사용자 ID를 보고 해당하는 N과 X_{N+1} 을 이용해서 도전(Challenge)값을 생성한다. 도전값은 N, 난수 R과 저장값 X_{N+1} 의 배타적 논리합 연산결과값($R \oplus X_{N+1}$) 및 $E_R(X_{N+1})$를 연쇄시킨 값($N \parallel R \oplus X_{N+1} \parallel E_R(X_{N+1})$)이다. 상기 도전값으로 사용자측으로 전송한다(43). 여기서 $E_R(X_{N+1})$은 난수 R을 키로하여 X_{N+1} 을 암호화하는 것을 말한다. 도전값을 수신한 사용자측은 사용자가 가지고 있던 비밀키 P를 도전값의 N 만큼 해쉬함수를 반복 수행해서 $H_N(P)$과 $H_{N+1}(P)$을 계산한다(44). 이것의 $H_{N+1}(P)$을 도전값중 난수 R과 저장값 X_{N+1} 의 배타적 논리합 연산결과값($R \oplus X_{N+1}$)의 배타적 논리합 연산을 수행하면 난수 R만 남게되고 이것을 R'로 옮긴다. 여기서 얻은 난수 R'를 이용해서 도전값의 $E_R(X_{N+1})$을 복호화하여 얻은 결과($D_{R'}(E_R(X_{N+1}))$)를 사용자 측에서 계산한 $H_{N+1}(P)$과 비교하여 일치하는지 여부를 판단한다. 일치하면, 서버측의 소유정보인 X_{N+1} 과 난수 R을 확인할 수 있어 서버가 인증된다(45). 사용자측은 다시 $H_N(P)$을 난수 R'로 암호화하여 응답(Response)값을 생성한후(46) 서버측으로 보낸다. 서버측에서는 이 응답 값을 수신하여 서버측에서 생성한 난수 R로 복호화한다. 여기서 얻어지는 $H_N(P)$에 해쉬함수를 수행하여 얻은 결과를 X_{N+1} 값과 비교하여 일치하면 사용자의 신분을 확인할 수 있게 되므로 사용자와 서버간의 상호 인증이 된다(47). 이 후에 서버측에서는 사용자에 관한 저장정보를 N+1에서 N으로, X_{N+1} 에서 $X_N=H_N(P)$로 갱신하면 모든 절차는 종료된다.

다시금 사용자측으로부터의 인증요구가 발생하면, 상기의 절차를 반복한다.

도5는 본 발명에 따른 인증 과정의 다른 실시예 상세 흐름도이다.

먼저, 사용자가 사용자 식별번호(ID)를 가지고 접근 요구를 서버측에 보낸다(51). 서버측에서는 난수 R을 생성한 후(52) 사용자의 ID를 보고 해당하는 N과 X_{N+1} 을 이용해서 도전(Challenge)값을 생성하여(53), 사용자측으로 전송한다. 도전값은 N, 난수 R과 저장값 X_{N+1} 의 배타적 논리합 연산결과값($R \oplus X_{N+1}$) 및 난수 R과 X_{N+1} 을 연쇄시켜 해쉬한 값($H(R \parallel X_{N+1})$)을 연쇄시킨 값($N \parallel R \oplus X_{N+1} \parallel H(R \parallel X_{N+1})$)이다. 도전값을 받은 사용자측은 사용자가 가지고 있던 비밀키 P를 도전값의 N만큼 해쉬함수를 반복 수행해서 $H_N(P)$와 $H_{N+1}(P)$를 계산한다(54). 이것의 $H_{N+1}(P)$를 도전값중 난수 R과 저장값 X_{N+1} 의 배타적 논리합 연산결과값($R \oplus X_{N+1}$)을 배타적 논리합 연산을 수행하면 난수 R만 남게 되고 이것을 R'로 옮긴다. 여기서 얻은 난수 R'와 $H_{N+1}(P)$값을 연쇄시킨 값의 해쉬함수를 수행하는 $H(R' \parallel H_{N+1}(P))$의 결과를 도전값중 $H(R \parallel X_{N+1})$과 비교하여 일치하는지 여부를 판단한다. 일치하면 서버측의 소유정보인 X_{N+1} 과 난수 R을 확인할 수 있으므로 서버 인증이 수행된다(55). 사용자측은 다시 X_N 즉, $H_N(P)$를 난수 R'와의 배타적 논리합 연산 수행 결과 ($H_N(P) \oplus R$)를 생성하여(56) 응답(Response)값으로 서버측에 보낸다. 서버측에서는 이 응답 값을 수신하여 서버측에서 생성한 난수 R과 사용자측으로부터 수신한 응답값을 배타적 논리합 연산하고 해쉬한 값 $H(R \oplus (H_N(P) \oplus R))$을 구한다. $H(R \oplus (H_N(P) \oplus R))$의 결과와 X_{N+1} 값을 비교하여 사용자 인증을 수행한다(57). 비교 결과가 일치하면 사용자의 신분을 확인할 수 있게 되므로 사용자와 서버간의 상호 인증이 완료된다. 이 후에 서버측에서는 사용자에 관한 저장정보를 N+1에서 N으로, X_{N+1} 에서 $X_N=H_N(P)$으로 갱신하면(58) 모든 절차는 끝나게 된다.

다시금 사용자측으로부터의 인증요구가 발생하면, 상기의 절차를 반복하면 된다.

이상에서 설명한 본 발명은 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

발명의 효과

상기한 바와 같은 본 발명은, 기존의 유닉스(UNIX) 패스워드 시스템의 패스워드 재시도 공격의 단점을 제거하고, 통신로상에서 평문으로 사용자의 신분을 인증하는 방법을 사용하는 위험과 서버로 위장하여 침입을 시도할 수 있는 S/KEY의 단점을 동시에 제거하여 인증의 안정성을 높일 수 있는 효과가 있다.

또한, 본 발명은 별도의 하드웨어 장치없이 일반적인 컴퓨터 시스템에 손쉽게 설치할 수 있고, 시스템의 성능에도 큰 영향을 미치지 않으므로, 사용자와 서버(Server)간의 안전한 상호인증이 필요한 모든 컴퓨터 통신망, 금융망 또는 PC 통신망 등에 적용하여 사용자 신분 위장을 방지할 수 있을 뿐만 아니라 서로간의 신뢰성을 확보할 수 있어서, 통신망 및 금융망등의 활성화를 유도하여 정보화를 촉진할 수 있다는 효과가 있다.

(57) 청구의 범위

청구항 1.
사용자와 서버간의 상호 신분 인증 방법에 있어서,

사용자와 서버가 초기화하여 상기 서버가 해쉬함수 수행 횟수인 N+1(여기서, N은 자연수) 및 해쉬 결과값 X_{N+1} 을 저장하는 제1 단계;

상기 사용자가 접근 요구를 전송하는 제2 단계;

상기 서버(Server)가 상기 접근 요구를 수신하여, 난수 R을 생성하고, 해쉬 수행 횟수 N, 난수 R과 상기 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값 및 상기 저장된 정보 X_{N+1} 의 암호화 값을 연쇄시킨 도전값($N \parallel R \oplus X_{N+1} \parallel E_R(X_{N+1})$)을 생성하여 사용자에게 전송하는 제3 단계;

상기 도전값을 수신한 상기 사용자가 자신의 비밀키 P와 해쉬함수를 이용하여 서버를 인증하기 위한 정보 $H_N(P)$, $H_{N+1}(P)$를 계산한후, $H_{N+1}(P)$과 상기 도전값 중 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값($R \oplus X_{N+1}$)의 배타적 논리합 연산을 수행하여 그 결과값 $R'(=(H_{N+1}(P) \oplus (R \oplus X_{N+1})))$를 구하고, 상기 R' 및 상기 도전값중 상기 저장된 정보 X_{N+1} 의 암호화 값 $E_R(X_{N+1})$을 복호화한 값 $D_{R'}(E_R(X_{N+1}))$을 $H_{N+1}(P)$ 값과 비교하여 일치하는지 여부를 판단하여 서버를 인증하는 제4 단계;

상기 $H_N(P)$을 상기 R'로 암호화하여 암호화값($E_{R'}(H_N(P))$)을 상기 서버에게 응답값으로 전송하는 제5 단계; 및

상기 서버는 상기 응답값 $E_{R'}(H_N(P))$을 수신한후, 상기 난수 R을 이용하여 복호화하고 복호화된 결과값을 해쉬함수의 입력으로하여 계산하여 결과를 상기 저장된 값 X_{N+1} 과 비교하여 사용자 신분을 확인하는 제6 단계

를 포함하는 사용자-서버간의 상호 신분 인증 방법.

청구항 2.
제1항에 있어서,

상기 제4 단계의 서버 인증 과정이 실패하면 상기 제2 단계부터 반복 수행하는 제7 단계를 더 포함하는 사용자-서버간의 상호 신분 인증 방법.

청구항 3.
제1항 또는 제2항에 있어서,

상기 제6 단계의 사용자 인증 과정이 실패하면 상기 제2 단계부터 반복 수행하는 제8 단계를 더 포함하는 사용자-서버간의 상호 신분 인증 방법.

청구항 4.
사용자와 서버간의 상호 신분 인증 방법에 있어서,

사용자와 서버가 초기화하여 상기 서버가 해쉬함수 수행 횟수인 N+1(여기서, N은 자연수) 및 해쉬 결과값 X_{N+1} 을 저장하는 제1 단계;

상기 사용자가 접근 요구를 전송하는 제2 단계;

상기 서버(Server)가 상기 접근 요구를 수신하여, 난수 R을 생성하고, 해쉬 수행 횟수 N, 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값 및 상기 난수 R과 상기 저장된 정보 X_{N+1} 의 연쇄값을 해쉬한 값을 연쇄시킨 도전값($N \parallel R \oplus X_{N+1} \parallel H(R \parallel X_{N+1})$)을 생성하여 사용자에게 전송하는 제3 단계;

상기 도전값을 수신한 상기 사용자가 자신의 비밀키 P와 해쉬함수를 이용하여 서버를 인증하기 위한 정보 $H_N(P)$, $H_{N+1}(P)$를 계산한후, $H_{N+1}(P)$과 상기 도전값 중 난수 R과 저장된 정보 X_{N+1} 의 배타적 논리합 연산 결과값($R \oplus X_{N+1}$)의 배타적 논리합 연산을 수행하여 그 결과값 $R'(=(H_{N+1}(P) \oplus (R \oplus X_{N+1})))$를 구하고, 상기 R'과 상기 $H_{N+1}(P)$를 해쉬한 값 $H(R' \parallel H_{N+1}(P))$을 도전값중 $H(R \parallel X_{N+1})$ 값과 비교하여 일치하는지 여부를 판단하여 서버를 인증하는 제4 단계;

상기 $H_N(P)$을 상기 R'과 논리합 연산하여 결과값($H_N(P) \oplus R'$)인 응답값을 상기 서버에게 전송하는 제5 단계; 및

상기 서버는 상기 응답값 ($H_N(P) \oplus R'$)을 수신한후, 상기 난수 R과 상기 응답값을 논리합 연산하여 해쉬한 값($H(R \oplus (H_N(P) \oplus R'))$)이 상기 저장된 값 X_{N+1} 과 일치하는지 여부를 비교하여 사용자 신분을 확인하는 제6 단계

를 포함하는 사용자-서버간의 상호 신분 인증 방법.

청구항 5.
제4항에 있어서,

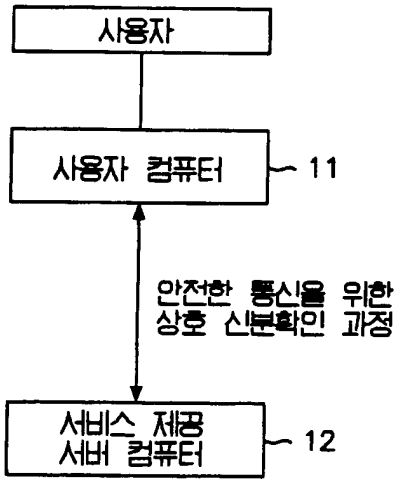
상기 제4 단계의 서버 인증 과정이 실패하면 상기 제2 단계부터 반복 수행하는 제7 단계를 더 포함하는 사용자-서버간의 상호 신분 인증 방법.

청구항 6.
제4항 또는 제5항에 있어서,

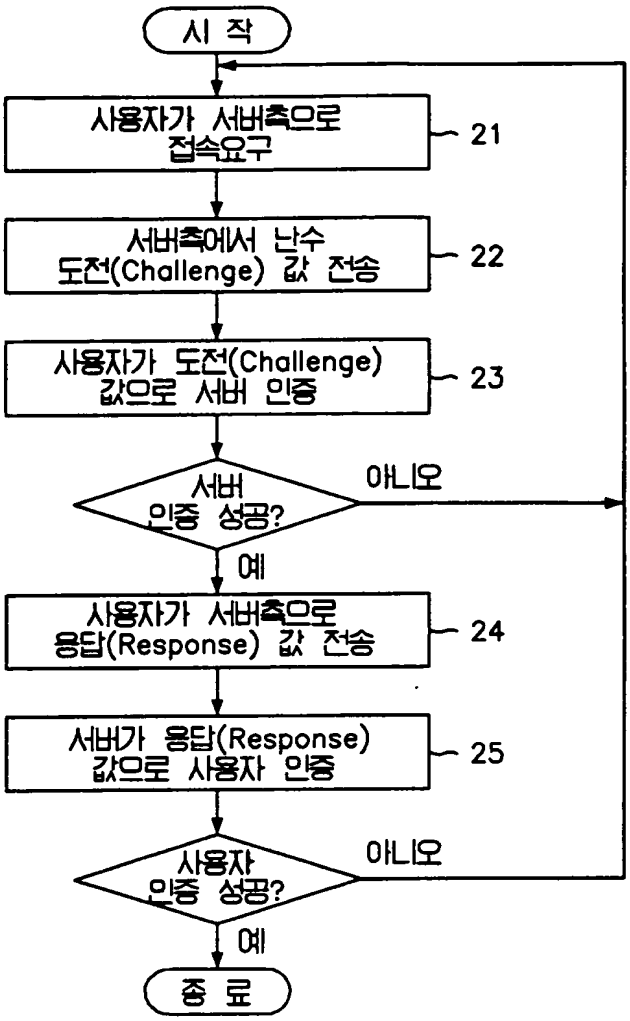
상기 제6 단계의 사용자 인증 과정이 실패하면 상기 제2 단계부터 반복수행하는 제8 단계를 더 포함하는 사용자-서버간의 상호 신분 인증 방법.

도면

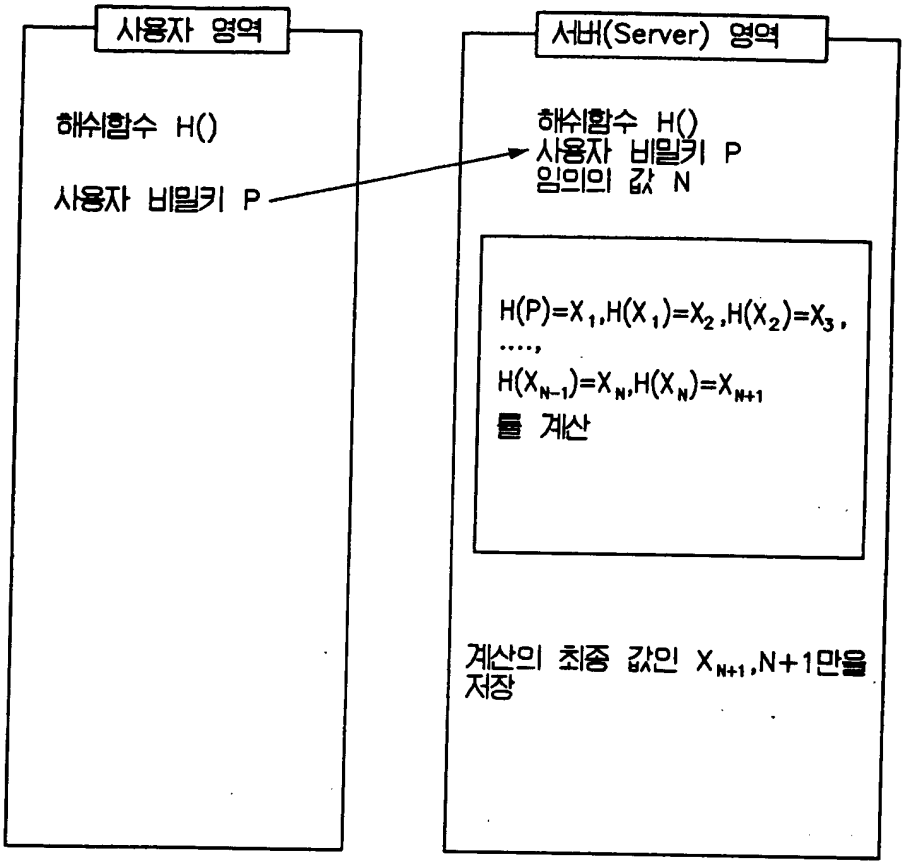
도면 1



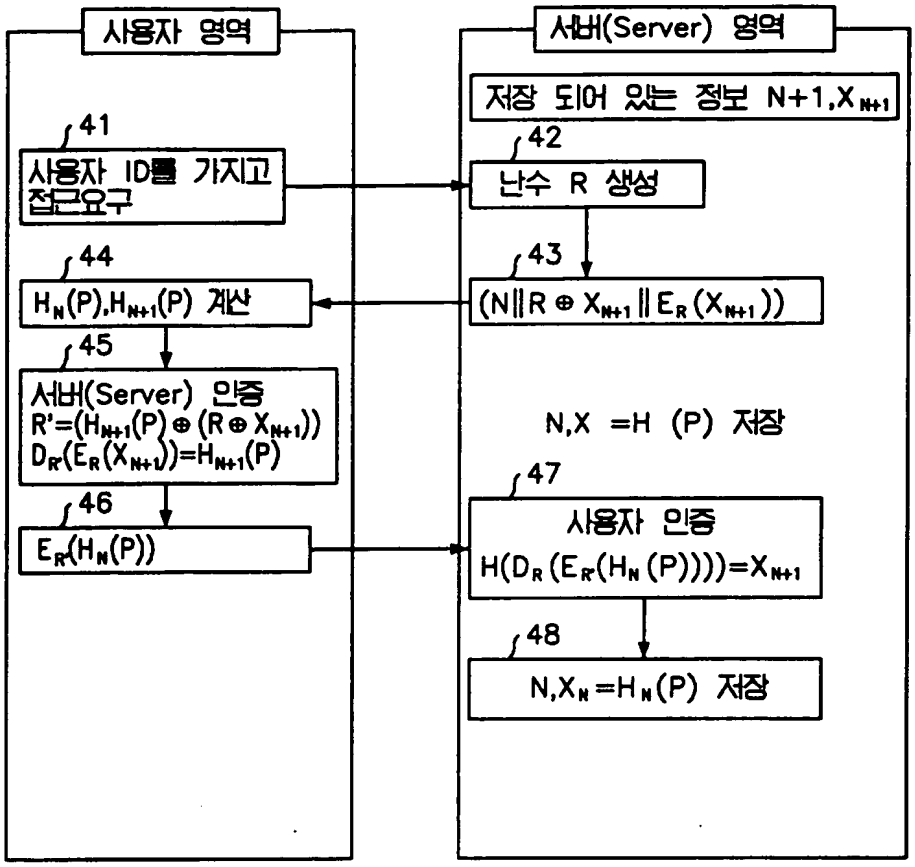
도면 2


시 작
사용자가 서버측으로 접속요구
21
서버측에서 난수 도전(Challenge) 값 전송
22
사용자가 도전(Challenge) 값으로 서버 인증
23
서버 인증 성공?
아니오
예
사용자가 서버측으로 응답(Response) 값 전송
24
서버가 응답(Response) 값으로 사용자 인증
25
사용자 인증 성공?
아니오
예
종 료

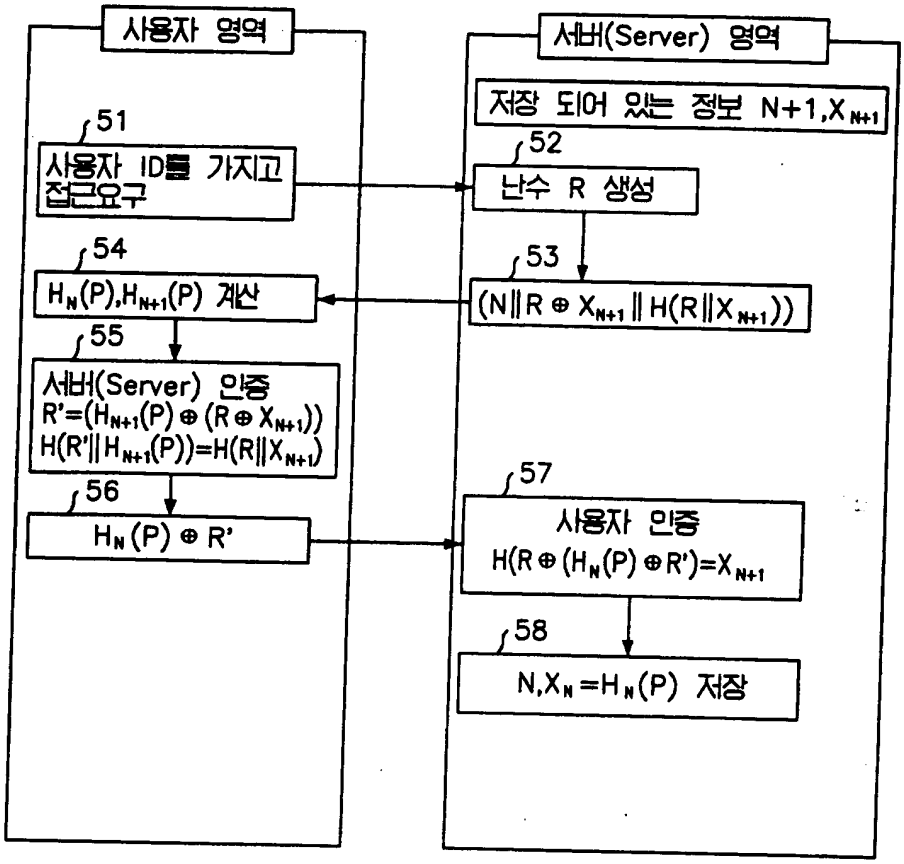
도면 3


사용자 영역
해쉬함수 H()
사용자 비밀키 P
서버(Server) 영역
해쉬함수 H()
사용자 비밀키 P
임의의 값 N
$H(P)=X_1, H(X_1)=X_2, H(X_2)=X_3,$
....,
$H(X_{N-1})=X_N, H(X_N)=X_{N+1}$
를 계산
계산의 최종 값인 $X_{N+1}, N+1$만을 저장

도면 4


사용자 영역
서버(Server) 영역
저장 되어 있는 정보 N+1,X N+1
41
사용자 ID를 가지고 접근요구
42
난수 R 생성
43
(N||R ⊕ X N+1 || E R (X N+1))
44
H N(P),H N+1(P) 계산
45
서버(Server) 인증
R'=(H N+1(P) ⊕ (R ⊕ X N+1))
D R'(E R(X N+1))=H N+1(P)
N,X =H (P) 저장
46
E R'(H N(P))
47
사용자 인증
H(D R (E R'(H N (P))))=X N+1
48
N,X N =H N(P) 저장

도면 5


사용자 영역
서버(Server) 영역
저장 되어 있는 정보 N+1,X N+1
51
사용자 ID를 가지고 접근요구
52
난수 R 생성
54
H N(P),H N+1(P) 계산
53
(N‖R ⊕ X N+1 ‖ H(R‖X N+1))
55
서버(Server) 인증
R'=(H N+1(P) ⊕ (R ⊕ X N+1))
H(R'‖H N+1(P))=H(R‖X N+1)
56
H N(P) ⊕ R'
57
사용자 인증
H(R ⊕ (H N(P) ⊕ R')=X N+1
58
N,X N =H N(P) 저장